

Geo-Indistinguishability: Differential Privacy for Location-Based Systems

Miguel E. Andrés
LIX, École Polytechnique
gueles@gmail.com

Nicolás E. Bordenabe
INRIA
LIX, École Polytechnique
nbordenabe@lix.polytechnique.fr

Konstantinos Chatzikokolakis
CNRS
LIX, École Polytechnique
kostas@chatzi.org

Catuscia Palamidessi
INRIA
LIX, École Polytechnique
catuscia@lix.polytechnique.fr

Abstract—The growing popularity of location-based systems, allowing unknown/untrusted servers to easily collect and process huge amounts of users’ information regarding their location, has recently started raising serious concerns about the privacy of this kind of sensitive information. In this paper we study geo-indistinguishability, a formal notion of privacy for location-based systems that protects the exact location of a user, while still allowing approximate information – typically needed to obtain a certain desired service – to be released.

Our privacy definition formalizes the intuitive notion of protecting the user’s location within a radius r with a level of privacy that depends on r . We present three equivalent characterizations of this notion, one of which corresponds to a generalized version of the well-known concept of *differential privacy*. Furthermore, we present a perturbation technique for achieving geo-indistinguishability by adding controlled random noise to the user’s location, drawn from a planar Laplace distribution. We demonstrate the applicability of our technique through two case studies: First, we show how to enhance applications for location-based services with privacy guarantees by implementing our technique on the client side of the application. Second, we show how to apply our technique to sanitize location-based sensible information collected by the US Census Bureau.

I. INTRODUCTION

In recent years, the increasing availability of location information about individuals has led to a growing use of systems that record and process location data, generally referred to as “location-based systems”. Such systems include (a) Location Based Services (LBSs), in which a user obtains, typically in real-time, a service related to his current location, as well as (b) location-data mining algorithms, used to determine, among others, points of interest and traffic patterns.

The use of LBSs, in particular, has been significantly increased by the growing popularity of mobile devices equipped with GPS chips, in combination with the increasing availability of wireless data connections. A recent study in the US shows that 46% of the adult population of the country owns a smartphone and, furthermore, that 74% of

those owners use LBSs [1]. Examples of LBSs include mapping applications (eg, Google Maps, Bing Maps), Points of Interest (POI) retrieval (eg, AroundMe, Localscope), coupon/discount providers (eg, GroupOn, Yowza), GPS navigation (eg, TomTom), and location-aware social networks (eg, Foursquare, OkCupid).

While location-based systems have demonstrated to provide enormous benefits to individuals and society, the growing exposure of users’ location information raises important privacy issues that are, unfortunately, often overlooked. On the one side, location information itself is commonly considered by individuals as sensitive. More importantly, location data can be easily linked to a variety of other information that an individual might wish to protect; by collecting and processing accurate location information on a regular basis, it is possible to infer an individual’s home or work location, sexual preferences, political views, religious inclinations, etc. In its extreme form, monitoring and control of an individual’s location has been even described as a form of slavery [2].

As a consequence, several notions of privacy for location-based systems have been proposed in the literature, many of them being variations of the k -anonymity concept, together with techniques to achieve these privacy guarantees. In Section II we give an overview of such existing notions of location privacy, and we discuss some of their shortcomings in relation to our motivating example of a real-time LBS. Aiming at addressing these shortcomings, we present a *novel formal privacy definition* for location-based systems, as well as a technique that allows users to disclose their location while satisfying the aforementioned privacy guarantee.

As a motivating example, we consider a user located in Paris who wishes to query an LBS provider for nearby restaurants in a private way, i.e. by disclosing some approximate information z instead of his exact location x . Note that, in contrast to various works in the literature, we assume that the user is interested in hiding his *location*, not his *identity*. In fact, the user might be willing to disclose his identity to the provider, in order to obtain personalized recommendations, or to participate in a social network. A crucial question then is, what kind of privacy does the user *expect* to have in this scenario? On the one hand, he does not expect to reveal his exact location but, on the other hand,

This work has been partially supported by the European Union Seventh Framework Programme under the grant agreement no. 295261 (MEALS), by the ANR-11-IS02-0002 project LOCALI, and by the INRIA Large Scale Initiative CAPPRIS. The work of Nicolás E. Bordenabe has been partially funded by the French Defense procurement Agency (DGA) by means of a PhD grant.



Figure 1. Geo-indistinguishability: different levels of privacy for each r

he wishes to obtain a service tailored to it. Thus, the user's requirement is that, by obtaining z , the provider should be able to infer x *approximately* but not *accurately*.

To capture this requirement, we use the notion of privacy *within a radius*. We fix a circle of radius r centered at the user's location x , and reason about the user's level of privacy within this radius. Roughly speaking, we say that the user enjoys ℓ -privacy *within r* if, by observing z , the provider's ability to infer x *among all points within the radius r* , does not increase (compared to the case when z is unknown) by more than a factor depending on ℓ . The idea is that ℓ is the (inverse of) user's *level* of privacy for that radius: the smaller ℓ is, the higher privacy the user enjoys (as it gets harder for the provider to detect the user's location among the locations within this circle).

Then, in order to allow the provider to learn x approximately but not accurately, we require a different level of privacy ℓ for each radius; in particular we require that ℓ decreases proportionally with r , which brings us to our (still informal) definition of *geo-indistinguishability*:

A mechanism satisfies ϵ -geo-indistinguishability iff for any radius $r > 0$, the user enjoys ϵr -privacy within r .

This definition requires that the user is protected within any radius r , but with a level $\ell(r) = \epsilon r$ that increases with the distance. Within a short radius, for instance $r = 1$ km, $\ell(r)$ is small, guaranteeing that the provider cannot infer the user's location within, say, the 7th arrondissement of Paris. On the other hand, privacy decreases for locations farther away from the user; taking for instance $r = 10.000$ km, $\ell(r)$ becomes very large, allowing the LBS provider to infer that with high probability the user is located in Paris instead of, say, London. The idea of different privacy levels for each radius is illustrated in Figure 1.

Geo-indistinguishability is formalized in Section III, where we present three equivalent characterizations offering alternative interpretations of our privacy guarantee. One of the characterizations corresponds to a generalized version of *differential privacy*, a well-known notion of privacy from the area of statistical databases [3]. This relationship emphasizes the fact that – like differential privacy – our notion abstracts

from the side information of the user, such as any prior probabilistic knowledge about the user's actual location.

Furthermore, we develop a mechanism to achieve geo-indistinguishability, by reporting a randomly selected point z , obtained by perturbing the user's location x . The inspiration for our mechanism comes from one of the most popular approaches for differential privacy, namely drawing noise from a Laplace distribution. This distribution, however, is one-dimensional, while a planar (two-dimensional) mechanism is required to generate noise for location values. Nevertheless, the Laplace distribution can be extended to the continuous plane obtaining, in this way, a distribution to draw noise for location values in a *geo-indistinguishable* fashion. Moreover, via a transformation to polar coordinates, we are also able to devise a simple and efficient method to draw points. However, as standard (digital) applications require a finite representation of locations, it is necessary to add a discretization step after randomly generating z , such operation degrades the level of privacy provided by the mechanism. Quantifying such degradation of privacy imposes several technical challenges; we show how to overcome them and how to adjust the privacy parameters of our mechanism in order to obtain a desired level of privacy.

We conclude our work by demonstrating the applicability of our approach through two case studies, one based on LBSs and the other on location-data mining. In the former case, we show that, by trading privacy for bandwidth usage, geo-indistinguishability can be obtained without degrading the utility of the information provided by the LBS. In the latter case, we show how to apply our technique to sanitize datasets containing geographical information. In particular, we show how to sanitize publicly available geographic information released by the US Census Bureau. Our experiments reveal that providing geo-indistinguishability to all users in the dataset (i.e., US inhabitants) does not significantly decrease the quality of the sanitized data (the degree of decrease being inversely proportional to the parameters ℓ and r of the privacy guarantee).

Road Map: In Section 2 we discuss notions of location privacy from the literature and point out their weaknesses and strengths. In Section 3 we formalize the notion of geo-indistinguishability in three equivalent ways. We then proceed to describe a mechanism that provides geo-indistinguishability in Section 4. In Sections 5 and 6 we demonstrate the applicability of our approach by case studies related to LBSs and Location-Data Mining, respectively. Section 7 discusses related work; Section 8 concludes. All proofs are in the appendix.

II. EXISTING NOTIONS OF LOCATION PRIVACY

In this section, we examine various notions of privacy from the literature, as well as techniques to achieve them. We consider the motivating example from the introduction, of a user in Paris wishing to find nearby restaurants with good

reviews. To achieve this goal, he uses a handheld device (eg. a smartphone) to query a public LBS provider. However, the user expects his location to be kept private: informally speaking, the information sent to the provider should not allow him to accurately infer the user's location. Our goal is to provide a *formal* notion of privacy that adequately captures the user's expected privacy. From the point of view of the employed mechanism for achieving privacy, we require a technique that can be performed in real-time by a handheld device such as a smartphone, without the need of any trusted anonymization party.

A. *k*-anonymity

The notion of *k*-anonymity is the most widely used definition of privacy for location-based systems in the literature. Many systems in this category ([4], [5], [6]) aim at protecting the user's *identity*, requiring that the attacker cannot infer which user is executing the query, among a set of *k* different users. Such systems are outside the scope of our problem, since we are interested in protecting the user's *location*.

On the other hand, *k*-anonymity has also been used to protect the user's location (sometimes called *l*-diversity in this context), requiring that it is indistinguishable among a set of *k* points (often required to share some semantic property). One way to achieve this is through the use of *dummy locations* ([7], [8]). This technique involves generating *k* - 1 properly selected dummy points, and performing *k* queries to the service provider, using the real and dummy locations. Another method for achieving *k*-anonymity is through *cloacking* ([9], [10], [11]). This involves creating a cloacking region that includes *k* points sharing some property of interest, and then querying the service provider for this cloacking region.

The main drawback of *k*-anonymity-based approaches in general is that a system cannot be proved to satisfy this notion unless assumptions are made about the attacker's side information. For example, dummy locations are only useful if they look equally likely to be the real location from the point of view of the attacker. Any side information that allows to rule out any of those points, as having low probability of being the real location, would immediately violate the definition.

Counter-measures are often employed to avoid this issue: for instance, [7] takes into account concepts such as ubiquity, congestion and uniformity for generating dummy points, in an effort to make them look realistic. Similarly, [11] takes into account the user's side information to construct a cloacking region. Such counter-measures have their own drawbacks: first, they complicate the employed techniques, also requiring additional data to be taken into account, making their application in real-time by a handheld device challenging. Moreover, the attacker's actual side information might simply be inconsistent with the assumptions being made.

As a result, notions that abstract from the attacker's side information, such as differential privacy, have been growing in popularity in recent years, compared to *k*-anonymity-based approaches.

B. Differential Privacy

Differential Privacy ([3]) is a notion of privacy from the area of statistical databases. Its goal is to protect an individual's data while publishing aggregate information about the database. Differential privacy requires that modifying a single user's data should have a negligible effect on the query outcome. More precisely, it requires that the probability that a query returns a value *v* when applied to a database *D*, compared to the probability to report the same value when applied to an *adjacent* database *D'* - meaning that *D*, *D'* differ in the value of a single individual - should be within a bound of e^ϵ . A typical way to achieve this notion is to add controlled random noise to the query output, for example drawn from a Laplace distribution. An advantage of this notion is that a mechanism can be shown to be differentially private independently from any side information that the attacker might possess.

Differential privacy has also been used in the context of location privacy. In [12], it is shown that a synthetic data generation technique can be used to publish statistical information about commuting patterns, while satisfying differential privacy. In [13], a quadtree spatial decomposition technique is used to ensure differential privacy in a database with location pattern mining capabilities.

As shown by the aforementioned works, differential privacy can be successfully applied in cases where aggregate information about several users is published. On the other hand, the nature of this notion makes it poorly suitable for applications in which a single individual is involved, such as our motivating scenario. The secret in this case is the location of a single user. Thus, differential privacy would require that any change in that location should have negligible effect on the published output, making it impossible to communicate any useful information to the service provider.

C. Transformation-based approaches

A number of approaches for location privacy are radically different from the ones mentioned so far. Instead of cloaking the user's location, they aim at making it completely invisible to the service provider. This is achieved by transforming all data to a different space, usually employing cryptographic techniques, so that they can be mapped back to spatial information only by the user ([14], [15]). The data stored in the provider, as well as the location send by the user are encrypted. Then, using techniques from Private Information Retrieval, the provider can return information about the encrypted location, without ever discovering which actual location it corresponds to.

A drawback of these techniques is that they are computationally demanding, making it difficult to implement them in a handheld device. Moreover, they require the provider's data to be encrypted, making it impossible to use popular providers, such as Google Maps, which have access to the real data.

III. GEO-INDISTINGUISHABILITY

In this section we formalize our notion of geo-indistinguishability. As already discussed in the introduction, the main idea behind this notion is that privacy is considered wrt a certain radius r , with a level that decreases proportionally with r . More precisely, a mechanism satisfies ϵ -geo-indistinguishability iff for any radius $r > 0$, the user enjoys ϵr -privacy within r . So far we kept the discussion on an informal level by avoiding to explicitly define what ℓ -privacy within r means. In the remaining of this section we formalize this notion in three different ways; all of them turn out to be equivalent, but they are all useful for understanding in depth the privacy guarantees provided by geo-indistinguishability.

Note that the parameter ϵ corresponds to the level of privacy at one unit of distance. For the user, a simple way to specify his privacy requirements is by a tuple (ℓ, r) , where r is the radius he is mostly concerned with and ℓ is the privacy level he wishes for that radius. In this case, it is sufficient to require ϵ -geo-indistinguishability for $\epsilon = \ell/r$; this will ensure a level of privacy ℓ within r , and a proportionally selected level for all other radii.

A. Probabilistic model

We introduce here the simple probabilistic model that is used in the rest of the paper. We start with a set \mathcal{X} of *points of interest*, typically the user's possible locations. Moreover, let \mathcal{Z} be a set of possible *reported values*, which in general can be arbitrary, although for our needs we consider \mathcal{Z} to also contain spatial points. In our operational scenario, the user is assumed to be at the location $x \in \mathcal{X}$. He then selects a point $z \in \mathcal{Z}$ which is made available to the attacker (for instance, it is reported to an untrusted service provider).

Probabilities come into place in two ways. First, the attacker might have side information about the user's location, knowing, for example, that he is likely to be visiting the Eiffel Tower, while unlikely to be swimming in the Seine river. Let X be the random variable giving the user's location (ranging over \mathcal{X}); the attacker's side information can be modelled by a *prior* distribution P_X for X , where $P_X(x)$ is the probability assigned to the location x .

Second, the selection of a point in \mathcal{Z} is itself probabilistic; for instance, z can be obtained by adding random noise to the actual location x (a technique used in Section IV). The probabilistic function for selecting a reported value based on the actual location is called a *mechanism*. Let Z be the random variable giving the reported point; a mechanism \mathcal{K} for selecting z is a function assigning to each location $x \in$

\mathcal{X} a probability distribution for Z , where $\mathcal{K}(x)(S)$ is the probability that the reported point belongs to the set $S \subseteq \mathcal{Z}$, when the user's location is x .¹ Together, P_X and \mathcal{K} induce a *joint* probability distribution P for X, Z , as $P(x, S) = P_X(x)\mathcal{K}(x)(S)$. Note that, by construction, $P(x) = P_X(x)$ and $P(S|x) = \mathcal{K}(x)(S)$.

B. First approach

We return to the issue of formalizing what ℓ -privacy within a radius r means. An intuitive way of doing so, is to compare the probabilities of different locations within r , after seeing a reported point in $S \subseteq \mathcal{Z}$ (note that we always consider sets of reported points, to allow for continuous distributions). Let $x, x' \in \mathcal{X}$, such that $d(x, x') \leq r$, where $d(\cdot, \cdot)$ denotes the Euclidean distance between points. Ideally, we would like to require that $P(x|S)/P(x'|S) \leq e^\ell$, meaning that for a small ℓ , the attacker assigns similar probabilities to the user being located in x or x' after observing S .

However, we would like our definition to hold for any side information that the attacker might have, meaning for all priors P_X . Intuitively, we cannot expect the above condition to hold for all priors, since two locations x, x' with very different prior probabilities (eg. the Eiffel Tower vs a location in the Seine) will also have different probabilities after the observation S . In other words, if $P(x)/P(x')$ is large, we cannot expect the corresponding fraction after observing S to be small. What we can expect, however, is that the two fractions, before and after the observation, are similar, meaning that S has limited effect to the probabilities assigned by the attacker. This brings us to our first formal definition of geo-indistinguishability:

Geo-indistinguishability-I: A mechanism satisfies ϵ -geo-indistinguishability iff for all priors P_X and all $S \subseteq \mathcal{Z}$:²

$$\frac{P(x|S)}{P(x'|S)} \leq e^{\epsilon r} \frac{P(x)}{P(x')} \quad \forall r > 0 \quad \forall x, x' : d(x, x') \leq r$$

C. Second approach

A second approach for defining privacy within a radius r , is to focus on a single location x and compare the probability of x before and after the observation. Ideally, we would like to require that $P(x|S)/P(x) \leq e^\ell$, meaning that for a small ℓ , the probability of x should not be affected by the observation S . However, this requirement is clearly too strong since some information is allowed to be leaked: a location in Paris might have negligible prior probability, since the user could be located anywhere in the world, while after the observation its probability is substantially increased.

¹For simplicity we assume X to be discrete, but allow Z to be continuous since we use continuous distributions in Section IV. Thus we consider probabilities of sets of points, implicitly assuming to be measurable.

²Note that for the sake of readability, we express the definitions in terms of fractions. To avoid issues with zero probabilities, we can write all definitions in flat form, i.e. $P(x|S)P(x') \leq e^{\epsilon r} P(x')P(x)$.

Remember, however, that we are interested in privacy *within the radius r* . Let $B_r(x)$ be the set of locations at distance at most r from x . Since we are interested in the attacker's capability of locating the user withing this radius, we condition all probabilities on the event $B_r(x)$. In other words, we reason about how accurately the attacker could infer a particular location x , if he already knew that the location was within $B_r(x)$. This brings us to our second definition of geo-indistinguishability:

Geo-indistinguishability-II: A mechanism satisfies ϵ -geo-indistinguishability iff for all priors P_X and all $S \subseteq \mathcal{Z}$:

$$\frac{P(x|S, B_r(x))}{P(x|B_r(x))} \leq e^{\epsilon r} \quad \forall r > 0 \forall x \in \mathcal{X}$$

D. Third approach

So far, we have considered the probability that the attacker assigns to locations before and after observing S , since comparing these probabilities is a natural way to quantify how much S helps the attacker. We now change our standpoint and consider instead the probabilities of observations, instead of locations. Intuitively, if two locations x, x' produce a reported value in S with similar probabilities, then S reveals little information about whether the actual location is x or x' . Thus, it is natural to require that $P(S|x)/P(S|x') \leq e^\ell$ for locations that lie within the radius r . This brings us to our final definition of geo-indistinguishability:³

Geo-indistinguishability-III: A mechanism satisfies ϵ -geo-indistinguishability iff for all $S \subseteq \mathcal{Z}$:

$$\frac{\mathcal{K}(x)(S)}{\mathcal{K}(x')(S)} \leq e^{\epsilon r} \quad \forall r > 0 \forall x, x' : d(x, x') \leq r$$

This definition requires that locations within close distance produce observations with similar probabilities. The farther away two locations are, the more different we allow the probabilities of producing S to be. This is very similar to the definition of differential privacy, which requires two databases that differ on a single individual to produce the same answer with similar probabilities, while databases that differ on many individuals are allowed to give an answer with different probabilities.

Note that differential privacy aims at completely protecting the value of an individual, requiring that arbitrary changes in his value should have negligible effect on the reported answer. In our scenario, however, such a requirement would be too strong, since the only information is the location of a single individual. Nevertheless, we are not interested in completely hiding the user's location, since some approximate information needs to be revealed in order to obtain the required service. This is achieved using a level of privacy that depends on the distance between locations.

³Note that since $P(S|x) = \mathcal{K}(x)(S)$, this definition can be given only in terms of \mathcal{K} , independently from the prior P_X .

Still, the connection between geo-indistinguishability and differential privacy is strong. In fact, the above definition is an instance of a generalized variant of differential privacy ([16], [17], [18]) which takes into account an arbitrary metric between secrets, where standard differential privacy corresponds to the so-called Hamming distance. In [16] the generalized definition is used to perform a compositional analysis of standard differential privacy for functional programs, while [17] uses metrics between individuals to define "fairness" in classification. In a companion paper [18], we study the generalized definition and show that different metrics provide different notions of privacy which can be suitable in various applications. This paper focuses on location-based systems and is, to our knowledge, the first work considering privacy under the Euclidean metric, which is a natural choice for spatial data.

Finally, we can show that the three definitions of geo-indistinguishability given in this section are simply different ways of expressing the same privacy requirement.

Theorem 3.1: Geo-indistinguishability-I, II, III coincide.

A note on the unit of measurement: Since the notion of distance between points is crucial for the definition of geo-indistinguishability, a natural question is: how is the definition affected by the unit in which distance is measured? Changing the unit causes all distances to be scaled; still, such a change should clearly not affect the privacy guarantees of a mechanism. The crucial point here is that, if r is a physical quantity expressed in some unit of measurement, then ϵ has to be expressed in the inverse unit, so that $\ell = \epsilon r$ is a pure number, thus it needs to be updated when the unit of measurement changes. For simplicity in the rest of this paper we omit the unit since the choice is orthogonal to our goals.

E. Protecting multiple locations

So far, we have assumed that the user has a single location that he wishes to communicate to a service provider in a private way (typically his current location). In practice, however, it is common for a user to have multiple points of interest, for instance a set of past locations or a set of locations he frequently visits. In this case, the user might wish to communicate to the provider some information that depends on all points, for instance the set of points itself, their centroid, etc. As in the case of a single location, privacy is still a requirement; the provider is allowed to obtain only approximate information about the locations, their exact value should be kept private. In this section, we discuss how ϵ -geo-indistinguishability extends to the case where the secret is a tuple of points $\mathbf{x} = (x_1, \dots, x_n)$.

Similarly to the case of a single point, the notion of distance is crucial for our definition. We define the distance between two tuples of points $\mathbf{x} = (x_1, \dots, x_n), \mathbf{x}' = (x'_1, \dots, x'_n)$ as:

$$d_\infty(\mathbf{x}, \mathbf{x}') = \max_i d(x_i, x'_i)$$

Intuitively, the choice of metric follows the idea of reasoning within a radius r : when $d_\infty(\mathbf{x}, \mathbf{x}') \leq r$, it means that all x_i, x'_i are within distance r from each other.

All definitions of this section can be then directly applied to the case of multiple points, by using d_∞ as the underlying metric. Enjoying ℓ -privacy within a radius r means that the observation can help the attacker to infer \mathbf{x} among all tuples at distance r (i.e. tuples having all points at distance r from the corresponding points of \mathbf{x}), by a factor of at most e^ℓ . All three definitions of geo-indistinguishability remain the same, the only change being the set of secrets and the distance between them.

Extending a mechanism to multiple points: A natural question then to ask is whether we can create a mechanism for tuples of points, by independently applying an existing mechanism \mathcal{K}_0 to each individual point, and report a tuple of values. Starting from a tuple $\mathbf{x} = (x_1, \dots, x_n)$, we independently apply \mathcal{K}_0 to each x_i obtaining a reported point z_i , and then report the tuple $\mathbf{z} = (z_1, \dots, z_n)$. Thus, the probability that the combined mechanism \mathcal{K} reports \mathbf{z} , starting from \mathbf{x} , is the product of the probabilities to obtain each point z_i , starting from the corresponding point x_i , i.e. $\mathcal{K}(\mathbf{x})(\mathbf{z}) = \prod_i \mathcal{K}_0(x_i)(z_i)$.⁴

The next question is what level of privacy does \mathcal{K} satisfy. For simplicity, consider a tuple of only two points (x_1, x_2) , and assume that \mathcal{K}_0 satisfies ϵ -geo-indistinguishability. At first look, one might expect the combined mechanism \mathcal{K} to also satisfy ϵ -geo-indistinguishability, however this is not the case. The problem is that the two points might be *correlated*, thus an observation about x_1 will reveal information about x_2 and vice versa. Consider, for instance, the extreme case in which $x_1 = x_2$. Having two observations about the same point reduces the level of privacy, thus we cannot expect the combined mechanism to provide the same level of privacy. Still, \mathcal{K} can be shown to satisfy privacy with a level that scales linearly with n :

Theorem 3.2: If \mathcal{K}_0 satisfies ϵ -geo-indistinguishability, then \mathcal{K} satisfies $n\epsilon$ -geo-indistinguishability.

Note that this issue is similar to the problem of composing queries in standard differential privacy. If the outcome of multiple queries is randomized by adding independent noise to each answer, then ϵ scales linearly with the number of queries. The reason is exactly that the answers are correlated, since they come from the same database.

Due to this scalability issue, the technique of independently applying a mechanism to each point is only useful when the number of points is small. Still, this is sufficient for some applications, such as the case study of Section V. Note also that this technique is by no means optimal: similarly to standard differential privacy ([19], [20]), better results could be achieved by adding noise to the whole tuple \mathbf{x} , instead

of each individual points. Developing such techniques for geo-indistinguishability is left as future work.

The case of uncorrelated points: In the previous paragraph we saw that when a mechanism is independently applied to multiple points, ϵ increases linearly with the number of points, that the points can be correlated. On the other hand, we are sometimes interested in applying a mechanism to uncorrelated points, that is points that are either selected independently from each other, or for which we can assume that the attacker has no information about their correlation. This can be captured by requiring that the probability to select x_i is independent from x_j and vice versa, that is $P(\mathbf{x}) = \prod_i P(x_i)$ (note that $P(x_i)$ is still arbitrary). Under this restriction, an observation about x_j does not intuitively reveal any information about x_i . Assuming that \mathcal{K}_i satisfies ϵ_i -geo-indistinguishability, it can be shown that the combined mechanism \mathcal{K} satisfies the same level of privacy wrt the *individual point* x_i , that is $P(\mathbf{z}|x_i) \leq e^{\epsilon_i} P(\mathbf{z}|x'_i)$ for all x_i, x'_i such that $d(x_i, x'_i) \leq r$. Note that the ϵ -geo-indistinguishability *might not* be satisfied for the tuple \mathbf{x} (we need to take $n\epsilon$ for this purpose). Still, assuming the lack of correlation, ϵ -geo-indistinguishability will be satisfied for each individual point x_i .

F. Comparison with standard differential privacy

As discussed in Section III-D, geo-indistinguishability is an instance of a generalized version of differential privacy, using the Euclidean metric to measure the distance between secrets. Thus, it is natural to examine how this notion compares to the one of standard differential privacy. As discussed in Section II, an advantage of geo-indistinguishability is that it can be applied to scenarios involving a single user, for which differential privacy is poorly suited. The comparison becomes more interesting in the case where secrets are tuples of n points, each corresponding to a different user. Note that we try to keep the discussion at a high level, focusing mainly on the privacy guarantees of each notion, and abstracting from the exact application.

Consider two mechanisms, \mathcal{K}_1 satisfying ϵ_1 -geo-indistinguishability and \mathcal{K}_2 satisfying ϵ_2 -differential privacy. Note that simply comparing ϵ_1, ϵ_2 is meaningless, since they refer to different definitions. To do a fair comparison, let $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{x}' = (x'_1, x_2, \dots, x_n)$, be two tuples differing only in the location of the first user (i.e. seen as databases, they are *adjacent*). We then consider the level of privacy that each mechanism provides *for those tuples*, which corresponds to how well the secret of the first user is protected. The privacy levels ℓ_1, ℓ_2 of $\mathcal{K}_1, \mathcal{K}_2$ respectively, for those tuples, is:

$$\ell_1 = \epsilon_1 d_\infty(\mathbf{x}, \mathbf{x}') = \epsilon_1 d(x_1, x'_1) \quad \ell_2 = \epsilon_2$$

in the sense that, for both mechanisms, the ratio $\mathcal{K}_i(\mathbf{x})(S)/\mathcal{K}_i(\mathbf{x}')(S)$ is bounded by e^{ℓ_i} for all observations

⁴For simplicity we consider probabilities of points here; a formal treatment of continuous mechanism would require to consider sets.

S . Thus, comparing the two mechanisms boils down to comparing ℓ_1, ℓ_2 , for various points x_1, x'_1 .

An important observation is that ℓ_2 is independent from the actual points x_1, x'_1 . This means that standard differential privacy protects all values in the same way; any secret value of a user is equally indistinguishable from any other. This is not the case for ℓ_1 , however, which depends on the actual points x_1, x'_1 , and more precisely on their distance. So, the level of protection depends on the secrets: the closer two points are the harder it is for the attacker to distinguish them.

Thus, for points far away from each other, ℓ_1 will be greater than ℓ_2 , so differential privacy offers better protection, while geo-indistinguishability becomes better in points close to each other, for which ℓ_1 is smaller than ℓ_2 . This behaviour becomes more important in cases where ϵ_2 is “weak”, which is often unavoidable in order to provide acceptable utility (see, for instance, Section VI). Intuitively, when ϵ_2 is large, then offering the same protection $\ell_2 = \epsilon_2$ for all points becomes a drawback. A privacy level that depends on the distance ensures that nearby points (which, in the case of location-based systems, need to be highly indistinguishable), will be adequately protected.

Finally, when comparing notions of privacy, one needs to also examine the loss of utility caused by the added noise. This highly depends on the application: differential privacy is suitable for publishing aggregate queries with *low sensitivity*, meaning that changes in a single individual have a relatively small effect on the outcome. On the other hand, location information often has high sensitivity. A trivial example is the case where we want to publish the complete tuple of points. But sensitivity can be high even for aggregate information: consider the case of publishing the centroid of 5 users located anywhere in the world. Modifying a single user can hugely affect their centroid, thus achieving differential privacy would require so much noise that the result would be useless. For geo-indistinguishability, on the other hand, one needs to consider the distance between points when computing the sensitivity. In the case of the centroid, a small (in terms of distance) change in the tuple has a small effect on the result, thus geo-indistinguishability can be achieved with much less noise.

IV. A MECHANISM FOR GEO-INDISTINGUISHABILITY

In this section we present a method to generate noise in a way that satisfies geo-indistinguishability. We model the location domain as the Euclidean plane equipped with the standard notion of Euclidean distance. This model can be considered a good approximation of the Earth surface when the area of interest is not “too large”.

For applications with digital interface the domain of interest is discrete, since the representation of the coordinates of the points is necessarily finite. However, it does not seem easy to devise an efficient mechanism for geo-indistinguishability that generates noise directly on a discrete

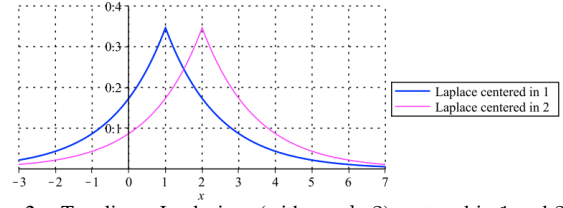


Figure 2. Two linear Laplacians (with $\epsilon = \ln 2$) centered in 1 and 2. The ratio between the two curves is at most $e^\epsilon = 2$ everywhere.

plane (we will come back to this point in Section IV-B). We therefore consider a different approach:

- (a) First, we define a geo-indistinguishable, continuous mechanism for the ideal case of the continuous plane.
- (b) Then, we discretized the mechanism by remapping each point generated according to (a) to the closest point in the discrete domain.

Furthermore, we may want to consider only a limited area. For instance if we are in an island, we may wish to report only locations in the land, not in the sea. Thus we may want to apply a third step:

- (c) If desirable, we may truncate the mechanism, so to report only points within the limits of the area of interest.

A. A geo-indistinguishable continuous mechanism

In this section we explore how to define a geo-indistinguishable mechanism on the continuous plane. This will constitute the basis of our method.

The idea is that whenever the actual location is $x_0 \in \mathbb{R}^2$, we report, instead, a point $x \in \mathbb{R}^2$ generated randomly according to the noise function. The property that we need to guarantee is that the probabilities of reporting a point in a certain (infinitesimal) area around x when the actual locations are x_0 and x'_0 respectively, should differ at most by a multiplicative factor $e^{-\epsilon d(x_0, x'_0)}$.

Intuitively, this property is achieved if the noise function is such that the probability of generating a point in the area around x decreases exponentially with the distance from the actual location x_0 . In a linear space this is exactly the behavior of the Laplace distribution, with probability density function (pdf) $\epsilon/2 e^{-\epsilon |x-\mu|}$ (where μ, ϵ are parameters). This distribution has been used in the literature to add noise to query results on statistical databases, with μ set to be the actual answer, and it can be shown to satisfy ϵ -differential privacy ([21]). Figure 2 illustrates the idea.

Of course we cannot use the standard Laplace distribution for our purposes, because it is defined on the line, while we need a distribution defined on the plane. Furthermore we need to use the (Euclidean) planar distance $d(x, \mu)$ instead of the linear distance $|x - \mu|$. Intuitively, however, just replacing $|x - \mu|$ by $d(x, \mu)$ in the Laplace’s pdf results in a natural extension of the Laplace distribution from one to

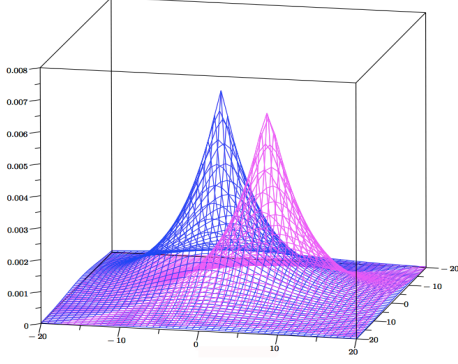


Figure 3. The pdf's of two planar Laplacians, centered in $(-2, -4)$ and in $(5, 3)$ respectively, with $\epsilon = 1/5$. The distance between the centers is $7\sqrt{2}$, and the ratio between the curves is at most $e^{7/5\sqrt{2}} \approx 7.24$ everywhere.

two dimensions⁵. We call *planar Laplacian* such extension.

The probability density function: Given the parameter $\epsilon \in \mathbb{R}^+$, and the actual location $x_0 \in \mathbb{R}^2$, the pdf of our noise mechanism, on any other point $x \in \mathbb{R}^2$, is:

$$D_\epsilon(x_0)(x) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d(x_0, x)} \quad (1)$$

where $\epsilon^2/2\pi$ is a normalization factor. Using a transformation in polar coordinates it is possible to show that the integral of this function over the whole \mathbb{R}^2 gives 1, which means that it is indeed the pdf of a probability distribution.

We call this function *planar Laplacian centered in x_0* . The corresponding distribution is illustrated by Figure 3. Note that the projection of a planar Laplacian on any vertical plane passing by the center gives a graph proportional to the one of a linear Laplacian (Figure 2). In Appendix B we show that the mechanism defined by a planar Laplacian satisfies ϵ -geo-indistinguishability.

Drawing a random point: We illustrate now how to draw a random point from the pdf defined in (1).

First of all, we note that the pdf of the planar Laplacian depends only on the distance from x_0 . It will be convenient, therefore, to transform the reference system into a system of polar coordinates with origin in x_0 . Intuitively, in this way the pdf will depend only on one variable, thus simplifying the drawing procedure.

So, given the pdf in (1), we consider the transformation into a system of polar coordinates (r, θ) where r is the radius and θ is the angle. A point x in Cartesian coordinates will be represented as a point (r, θ) in the new system, where r is the distance of x from x_0 , and θ is the angle that the line $x x_0$ forms with respect to the horizontal axis of the Cartesian system. Following the standard transformation method, the

⁵In the literature there are various proposals for the extension of the Laplace distribution to higher dimensions. These are called *multivariate Laplacians*. In general *multivariate* means that it involves $k \geq 1$ random variables. The particular cases of $k = 1$ and $k = 2$ are called *univariate* and *bivariate* respectively. Our definition corresponds to a particular instance of the extension investigated in [22]. The same instance has been adopted also in [23].

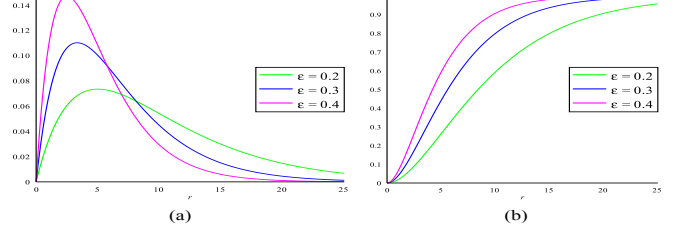


Figure 4. Gamma distribution: pdf and cdf for various values of ϵ .

pdf of the *polar Laplacian* centered in the origin (x_0) is:

$$D_\epsilon(r, \theta) = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r} \quad (2)$$

We note now that the polar Laplacian defined above enjoys a property that is very convenient for drawing in an efficient way: *the two random variables that represent the radius and the angle are independent*. Namely, the pdf can be expressed as the product of the two marginals. In fact, let us denote these two random variables by R (the radius) and Θ (the angle). The two marginals are:

$$D_{\epsilon, R}(r) = \int_0^{2\pi} D_\epsilon(r, \theta) d\theta = \epsilon^2 r e^{-\epsilon r}$$

$$D_{\epsilon, \Theta}(\theta) = \int_0^\infty D_\epsilon(r, \theta) dr = \frac{1}{2\pi}$$

Hence we have $D_\epsilon(r, \theta) = D_{\epsilon, R}(r) D_{\epsilon, \Theta}(\theta)$.

Note that $D_{\epsilon, R}(r)$ corresponds to the pdf of the *gamma distribution* with shape 2 and scale $1/\epsilon$. Figure 4 shows the graph of this function for various values of ϵ .

It may come as a surprise that this graph differs significantly from those in Figures 2 and 3, and in particular, that it does not have its maximum in the origin. Remember, however, that the graph in Figure 4(a) represents a pdf *in polar coordinates*. More precisely, $D_{\epsilon, R}(r)$ represents the probability that the random point is located in the circular crown centered in the origin and delimited by r and $r + dr$. The area of this crown is proportional to r , hence when r is close to 0 also the probability is close to 0. As r increases the probability increases, until the factor $e^{-\epsilon r}$ takes over. For r approaching infinity, the factor $e^{-\epsilon r}$ approaches 0, and dominates over r , hence the probability approaches 0 again.

Thanks to the fact that R and Θ are independent, in order to draw a point (r, θ) from $D_\epsilon(r, \theta)$ it is sufficient to draw separately r and θ from $D_{\epsilon, R}(r)$ and $D_{\epsilon, \Theta}(\theta)$ respectively.

Since $D_{\epsilon, \Theta}(\theta)$ is constant, drawing θ is easy: it is sufficient to generate θ as a random number in the interval $[0, 2\pi)$ with uniform distribution.

We now show how to draw r . Following standard lines, we consider the cumulative distribution function (cdf) $C_\epsilon(r)$:

$$C_\epsilon(r) = \int_0^r D_{\epsilon, R}(\rho) d\rho = 1 - (1 + \epsilon r) e^{-\epsilon r}$$

Intuitively, $C_\epsilon(r)$ (Fig 4(b)) represents the probability that the radius of the random point falls between 0 and r . Finally,

Drawing a point (r, θ) from the polar Laplacian

1. draw θ uniformly in $[0, 2\pi)$
2. draw z uniformly in $[0, 1)$ and set $r = C_\epsilon^{-1}(z)$

Figure 5. Method to generate Laplacian noise.

we generate a random number z with uniform probability in the interval $[0, 1)$, and we set $r = C_\epsilon^{-1}(z)$. Note that

$$C_\epsilon^{-1}(z) = -\frac{1}{\epsilon} \left(W_{-1} \left(\frac{z-1}{e} \right) + 1 \right)$$

where W_{-1} is the Lambert W function (the -1 branch), which can be computed efficiently.

Given a “universal” Cartesian reference system and the actual location $x_0 = (s, t)$ in this system, if we could work in the “ideal” continuous plane, then we would just need to generate the noise (r, θ) as specified above, and then reports the point $x = (s + r \cos \theta, t + r \sin \theta)$. In practice however there is always some discretization involved, because (a) computers have finite precision, and (b) (more important) the coordinates of the “universal reference system” will have a finite representation, typically using only a few decimal digits. The discretization of our method, and its properties, constitute the subject of next section.

B. Discretization

In practical applications locations are typically represented by means of discrete coordinates. For instance, latitude and longitude up to some decimal of precision. Thus we study here how to define an approximation of the Laplace distribution on a grid \mathcal{G} of discrete Cartesian coordinates. Again, the property that we need to preserve is that the probability of generating a point x in the grid decreases exponentially with the distance from the actual location x_0 .

Before we start illustrating our method, we wish to explain why we did not adopt the following approach, which seems the most natural: In the univariate case, the discrete approximation of the Laplace distribution is the double geometric probability distribution $\lambda e^{-\epsilon |x-x_0|}$, where $x \in \mathbb{Z}$ and λ is a normalization factor. This probability function can be visualized as a symmetric series of “steps” exponentially decreasing with the (discrete) distance from x_0 . The obvious extension to the bivariate (discrete) case would then be the probability distribution $K(x_0)(x) = \lambda' e^{-\epsilon d(x_0, x)}$ where λ' is a suitable normalization factor.

Unfortunately, there does not seem to be an efficient way to draw points according to the above distribution. For this reason we propose a different approach, that can be summarized as follows. Given the actual location x_0 , we report the point x in \mathcal{G} obtained in the following way:

- (a) first, we draw a point (r, θ) from the polar Laplacian centered in x_0 (see (2)), as described in Figure 5,
- (b) then, we remap (r, θ) to the closest point x on \mathcal{G} .

We will denote by $K_\epsilon : \mathcal{G} \rightarrow \mathcal{P}(\mathcal{G})$ the above mechanism. In summary, $K_\epsilon(x_0)(x)$ represents the probability of reporting the point x when the actual point is x_0 .

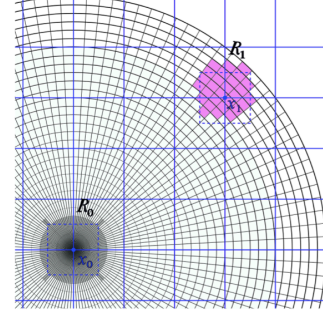


Figure 6. Remapping the points in polar coordinates to points in the grid.

It is not obvious that the discretization preserves geo-indistinguishability, due to the following problem: In principle, each point x in \mathcal{G} should gather the probability of the set of points for which x is the closest point in \mathcal{G} , namely

$$R(x) = \{y \in \mathbb{R}^2 \mid \forall x' \in \mathcal{G}. d(y, x') \leq d(y, x)\}$$

However, due to the finite precision of the machine, the noise generated according to (a) is already discretized in accordance with the polar system. Let \mathcal{W} denote the discrete set of points actually generated in (a). Each of those points (r, θ) is drawn with the probability of the area between r , $r + \delta_r$, θ and $\theta + \delta_\theta$, where δ_r and δ_θ denote the precision of the machine in representing the radius and the angle respectively. Hence, step (b) generates a point x in \mathcal{G} with the probability of the set $R_{\mathcal{W}}(x) = R(x) \cap \mathcal{W}$. This introduces some irregularity in the mechanism, because the scaly region associated to $R_{\mathcal{W}}(x)$ has a different shape and area depending on the position of x relatively to x_0 .

Figure 6 illustrates the situation. The Cartesian grid constituted by blue horizontal and vertical lines represents \mathcal{G} . The polar grid constituted by black circles and radial lines represent \mathcal{W} . The two dashed rectangles around the points x_0 and x_1 represent $R(x_0)$ and $R(x_1)$. The regions R_0 and R_1 colored in grey and magenta correspond to $R_{\mathcal{W}}(x_0)$ and $R_{\mathcal{W}}(x_1)$ respectively. Note that R_0 and R_1 have different shapes and areas, for instance R_0 is larger than R_1 .

In the next paragraph we show that, despite of the above problem, we can still obtain ϵ -geo-indistinguishability, at the price of replacing the ϵ of K_ϵ by a smaller ϵ' .

Geo-indistinguishability of the discretized mechanism:

We now analyze the privacy guarantees provided by our discretized mechanism. We show that the discretization preserves geo-indistinguishability, at the price of introducing some additional noise. More precisely we show that $K_{\epsilon'}$ satisfies ϵ -geo-indistinguishability, within a range r_{\max} , provided that ϵ' is chosen in a suitable way that depends on ϵ , on the length of the step units of \mathcal{G} , and on the precision of the machine.

For the sake of generality we do not require the step units along the two dimensions of \mathcal{G} to be equal. We will call them *grid units*, and will denote by u and v the smaller and the larger unit, respectively. We recall that δ_θ and δ_r

denote the precision of the machine in representing θ and r , respectively. We assume that $\delta_r \leq r_{\max} \delta_\theta$. The following theorem states the geo-indistinguishability guarantees provided by our mechanism.

Theorem 4.1: Assume $r_{\max} < u/\delta_\theta$, and let $q = u/r_{\max} \delta_\theta$. Let $\epsilon, \epsilon' \in \mathbb{R}^+$ such that

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \leq \epsilon$$

Then $K_{\epsilon'}$ provides ϵ -geo-indistinguishability within the range of r_{\max} . Namely, if $d(x_0, x), d(x'_0, x) \leq r_{\max}$ then:

$$K_{\epsilon'}(x_0)(x) \leq e^{\epsilon d(x_0, x'_0)} K_{\epsilon'}(x'_0)(x).$$

The difference between ϵ' and ϵ , in Theorem 4.1, represents the extra noise that we need to add in order to compensate the effect of discretization. Note that r_{\max} , which determines the area in which ϵ -geo-indistinguishability is guaranteed, must be chosen in such a way that $q > 2e^{\epsilon' u}$. Furthermore there is a trade-off between ϵ' and r_{\max} : If we want ϵ' to be close to ϵ then we need q to be large. Depending on the precision, this may or may not imply a serious limit on r_{\max} . Vice versa, if we want r_{\max} to be large then, depending on the precision, ϵ' may need to be significantly smaller than ϵ , and furthermore we may have a constraint on the minimum possible value for ϵ , which means that we may not have the possibility of achieving an arbitrary level of geo-indistinguishability.

Figure 7 shows the relation between ϵ and the maximal ϵ' satisfying the condition of Theorem 4.1. In all cases the grid unit is $u = 3 \cdot 10^{-3}$ Km = 3 m, and the other parameters are as follows:

- The green line corresponds to $q = 3 \cdot 10^9$. For instance this value can be obtained with double precision (16 significant digits, i.e., $\delta_\theta = 10^{-16}$) and $r_{\max} = 10^4$ Km. In the case of double precision, even for much larger values of r_{\max} (up to about 10^6 Km) ϵ' coincides with ϵ .
- The magenta line corresponds to $q = 3 \cdot 10^2$. This value can be obtained with single precision (7 significant digits, i.e., $\delta_\theta = 10^{-7}$) and $r_{\max} = 10^2$ Km. In this case we cannot go much higher for r_{\max} without ϵ' diverging dramatically from ϵ . Furthermore, the smallest possible value for ϵ is about 4.5, which means that at most we can ensure 4.5-geo-indistinguishability.
- The blue line corresponds to $q = 3 \cdot 10^3$, which can still be obtained with single precision at the price of reducing previous r_{\max} by a factor 10 ($r_{\max} = 10$ Km). Alternatively we could obtain this value by increasing both the precision and r_{\max} : For instance, with an intermediate precision of 9 significant digits ($\delta_\theta = 10^{-9}$) and $r_{\max} = 10^3$ Km.

Note that in Theorem 4.1 the restriction about r_{\max} is crucial. Namely, ϵ -geo-indistinguishability does not hold for arbitrary distances for any finite ϵ . Intuitively, this is because

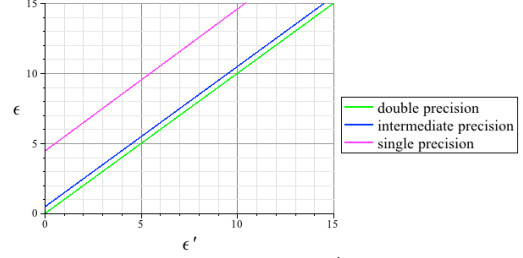


Figure 7. The relation between ϵ and ϵ' for various precisions.

the step units of \mathcal{W} (see Figure 6) become larger with the distance r from x_0 . The step units of \mathcal{G} , on the other hand, remain the same. When the steps in \mathcal{W} become larger than those of \mathcal{G} , some x 's have an empty $R_{\mathcal{W}}(x)$. Therefore when x is far away from x_0 its probability may or may not be 0, depending on the position of x_0 in \mathcal{G} , which means that geo-indistinguishability cannot be satisfied.

On the other hand, the restriction on r_{\max} is not a strong limitation, because the distribution decreases exponentially with r , and r_{\max} is usually large, hence the points with distance $r > r_{\max}$ have negligible probability.

C. Truncation

In practical applications we are typically interested in locations within a certain region. The Laplacian mechanisms described in previous sections, however, has the potential to generate points everywhere in the plane. If the user knows that the actual location is situated within a certain region, it seems desirable that the reported location lies within the same region as well, or at least not too far apart. To this purpose we propose a variant of the discrete Laplacian described in previous section, which generates points only within a specified region.

We assume that the specified region \mathcal{A} of acceptable report points is a circle centered in o , and diameter $\text{diam}(\mathcal{A})$. Our mechanism works like the discretized Laplacian of previous section, with the difference that, whenever the point generated in step (a) lies outside \mathcal{A} , we remap it to the closest point in $\mathcal{A} \cap \mathcal{G}$ (which necessarily will be on the perimeter of \mathcal{A} , modulo discretization).

Let us denote by $K_{\epsilon'}^T$ the truncated variant of the mechanism $K_{\epsilon'}$ described in previous section. The type is: $K_{\epsilon'}^T : \mathcal{A} \rightarrow \mathcal{P}(\mathcal{A} \cap \mathcal{G})$ and the drawing is described by the following procedure. Given the actual location $x_0 \in \mathcal{A}$:

- first, draw a point (r, θ) from the polar Laplacian centered on x_0 , as explained in previous section,
- then, remap (r, θ) to the closest point x on $\mathcal{A} \cap \mathcal{G}$.

Intuitively, K^T behaves like K except when the region $R(x)$ is on the border of \mathcal{A} . In this case, the probability on x is given not only by the probability of the points in $R_{\mathcal{W}}(x)$, but also by the probability of the part of the cone determined by o and $R(x)$ which lies outside \mathcal{A} .

We are now going to show that this new method satisfies geo-indistinguishability on all \mathcal{A} , provided that r_{\max} is not

smaller than $\text{diam}(\mathcal{A})$.

Theorem 4.2: Let r_{\max} , ϵ and ϵ' satisfy the premise of Theorem 4.1. If $r_{\max} \geq \text{diam}(\mathcal{A})$, then $K_{\epsilon'}^T$ provides ϵ -geo-indistinguishability within \mathcal{A} .

In the following we generally assume $\mathcal{A} = r_{\max}$.

V. ENHANCING LBSs WITH PRIVACY

In this section we present a case study of our privacy mechanism in the context of LBSs. In particular we show how to enhance LBS applications with privacy guarantees while still providing a high quality service to their users.

A. Geo-indistinguishability for POI retrieval LBSs

Let us start by describing how geo-indistinguishability can be used to specify a subtle notion of privacy for LBS applications. For that purpose, we first delineate the architecture of LBS applications that we consider in this work. We assume a simple client-server architecture where users communicate via a trusted mobile application (the client – typically installed in a smart-phone) with an unknown/untrusted LBS provider (the server – typically running on the cloud). Hence, our approach does not rely on trusted third-party servers (in contrast to several solutions proposed in the literature). Additionally, since this work focuses on the potential harm incurred to users by conferring their location to a LBS, we assume that users only communicate location information to the provider (although typically more information, such as user ID and network address, is transmitted). Figure 8 illustrates the LBS setting that we consider in this work.

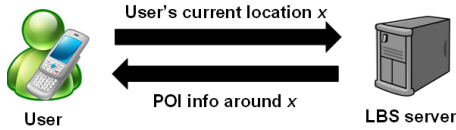


Figure 8. LBS architecture

For illustration purposes, in this section we will focus on LBSs applications providing POI information. However, most of the discussion and techniques presented in the following, hold for a broader family of LBS applications (some of which we mention explicitly below).

Coming back to our running example, we now study how geo-indistinguishability can help to provide privacy guarantees to the user visiting Paris. More precisely, let us assume that the user is sitting at Café Les Deux Magots and wishes to obtain information about nearby restaurants without revealing to a potential attacker (the LBS provider in this case) his exact location. However, as discussed before, in order to obtain accurate information from the LBS provider, the user is willing to reveal some approximate information.

This privacy guarantee can be captured by our notion of ϵ -geo indistinguishability. Letting the user specify his desired level of privacy, say $\ell = \ln(4)$ within $r = 0.2$ km (and decreasing proportionally for larger distances), $\ln(4)/0.2$ -geo-indistinguishability guarantees the user that by using the

Sanitizing Algorithm for a Location – NoisyPt

Input: x // point to sanitize
 ϵ // privacy parameter
 $u, v, \delta_\theta, \delta_r$ // precision parameters – Section IV-B
 \mathcal{A} // region of acceptable locations – Section IV-B

Output: Sanitized version x' of input x

1. $q = u/\mathcal{A}\delta_\theta$;
2. $\epsilon' = \text{safe_}\epsilon(\epsilon, u, v, q)$; // Theorem 4.2
3. Draw angle $\theta \sim \text{Uniform}(2\pi)$; // Figure 5
4. Draw radius $r \sim \text{gamma}(2, 1/\epsilon')$; // Figure 5
5. $x' = \text{Pt}(x, \rho, \theta)$; // sanitized location
6. **if** $x' \notin \mathcal{A}$ **then** $x' = \text{closestPt}(\mathcal{A}, x, \rho, \theta)$; // truncation
7. **return** x' ;

Figure 9. Our sanitizing algorithm for a location.

LBS application (and thus revealing his approximate location), the LBS provider cannot infer his real location (at least not with probability 4 times higher than without revealing his location) among all locations within 200 meters.

B. Privately Retrieving POI information from a LBS

We now proceed to describe how to enhance LBS applications with geo-indistinguishability guarantees. In the following we distinguish between *mildly-location-sensitive* and *highly-location-sensitive* LBS applications.

The former category corresponds to LBS applications offering a service that does not heavily rely on the precision of the location information provided by the user. Examples of such applications are weather forecast applications (forecast information for an approximate location is typically as good as forecast information for an exact location), location-aware advertising/offers (eg, shops offering discounts typically care about users being nearby – rather than their exact location), and a number of LBS applications for POI retrieval (eg, retrieving nearby cheap gas stations or nearby tourist sites when visiting a city). Enhancing this kind of LBSs with geo-indistinguishability privacy guarantees is relatively straightforward. It requires to implement the location perturbation mechanism presented in Section IV on the client party of the LBS application and then report the sanitized location (instead of the real location) to the LBS server party. We note that this simple modification does not impose a significant computation overlay on the client side nor extra bandwidth usage. Figure 9 delineates a location sanitizing algorithm based on the techniques described in Section IV-B.

For highly-location-sensitive LBS applications, on the other hand, the quality of the service provided by LBSs highly depends on the precision of the location information submitted by the user. Our running example lies within this category. For the user sitting at Café Les Deux Magots, information about restaurants nearby Champs Élysées is considerably less valuable than information about restaurants around his location. Enhancing this kind of LBS applications with privacy guarantees is considerably more challenging. In the following we describe how to enhance this kind

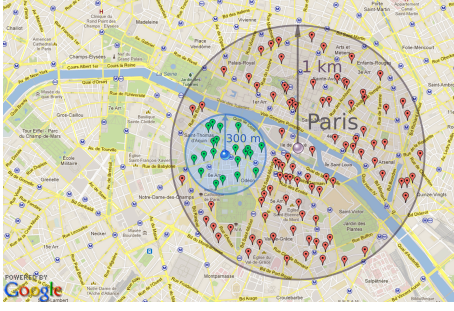


Figure 10. Retrieval information situation for private LBS

of LBS applications with privacy guarantees while still providing a high quality service. Our approach requires three modifications to the standard LBS architecture:

- 1) The algorithm illustrated in Figure 9 should be implemented on the client application in order to report to the LBS server party the user's approximate location z rather than his real location x .
- 2) Due to the fact that the information retrieved from the server is about POI nearby z , the area of POI information retrieval should be increased. In this way, if the user wishes to obtain information about POI within, say, 300 meters of x , the client application should request information about POI within, say, 1 km of z . Figure 10. illustrates the situation for our running example. The user's current location x is at café Les Deux Magots and the reported approximate location z submitted by the client application is at about 500 meters from x . We will refer to the circle centered at x with 300 meters radius as *area of interest* (of the user) and to the circle centered at z with 1 km radius as *area of retrieval*.
- 3) Finally, the client application should filter the retrieved POI information (depicted by the pins within the area of retrieval in Figure 10) in order to provide to the user with the desired information (depicted by pins within the user's area of interest in Figure 10).

The resulting client-server interaction is shown in Fig 11.

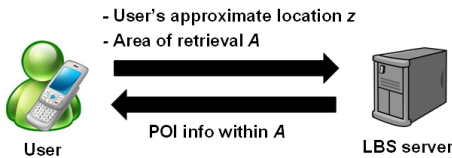


Figure 11. LBS architecture

Clearly, for our approach it is crucial that the area of interest is fully contained in the area of retrieval (otherwise the information expected by the user might not be fully retrieved from the server). However, the latter depends on a randomly generated location, hence such condition cannot be guaranteed (at least not with probability 1). Note that the client application could dynamically adjust the area of

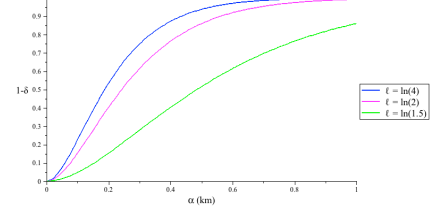


Figure 12. (α, δ) -usefulness for $r = 0.2$ and various values of ℓ .

retrieval in order to ensure that it always contains the area of interest. However, this approach would jeopardize the privacy guarantees: on the one hand, the size of the area of retrieval would leak information about the user's real location and, on the other hand, the LBS provider would know with certainty that the user is located within the retrieval area. Therefore, in order to provide geo-indistinguishability in this setting, the area of retrieval should be defined *independently* from the randomly generated location.

Our approach consists on *statically* defining the area of retrieval as a function of the security parameters (ℓ and r) and of the area of interest. Our goal is to define an area of retrieval as small as possible (in order to avoid retrieving unnecessary information and, consequently, unnecessary bandwidth usage) in a way that the area of interest is contained in it with probability as high as possible. Since such goal highly depends on the *accuracy* of the mechanism generating the approximate location (ie, on how close the generated location and the real location are to each other) before presenting our solution we need to introduce the notion of accuracy for data sanitation mechanisms.

C. Accuracy for location perturbation mechanisms

As it is standard for privacy enhancing mechanisms based on data perturbation (eg, the Laplacian mechanism providing standard differential privacy [21]), the aim of our mechanism is to provide accurate (location) information in a private way (ie, while satisfying geo-indistinguishability).

In order to evaluate the accuracy of our mechanism, we use (α, δ) -usefulness [19], a well-known concept from the literature (adapted to our location setting) that aims at assessing the accuracy of the approximate information generated by a mechanism.

A location perturbation mechanism \mathcal{K} is (α, δ) -useful if for every location x , with probability at least $1 - \delta$, the reported location $z = \mathcal{K}(x)$ satisfies $d(x, z) \leq \alpha$. In other words, a (α, δ) -useful mechanism generates approximate locations z within distance α of the exact location x with probability at least $1 - \delta$. In the case of our mechanism, δ can be computed using the cdf of the Gamma distribution from which the radius is drawn. Figure 12 illustrates how our mechanism behaves with respect to (α, δ) -usefulness when providing ϵ -geo-indistinguishability for $r = 0.2$ (as in our running example) and several values of ℓ .

It follows from the information in Figure 12, that a mechanism providing the privacy guarantees specified in our

running example (ϵ -geo-indistinguishability, with $\ell = \ln(4)$ and $r = 0.2$) generates an approximate location z falling within 1 km of the user's location x with probability 0.99, falling within 690 meters with probability 0.95, falling within 560 meters with probability 0.9, and falling within 390 meters with probability 0.75.

We now have all the necessary ingredients to define an area of retrieval containing the area of interest with a given probability. Note that an area of retrieval with radius, say, r_A contains the area of interest with radius say, r_I , with probability at least $1 - \delta$ if the mechanism used to generate the reported location is (α, δ) -useful, for an $\alpha \leq r_A - r_I$.

Therefore, by setting r_A to 1 km in our running example and since our mechanism is $(0.69, 0.05)$ -useful, it is guaranteed that the retrieval area contains the area of interest with probability at least 0.95.

D. Further challenges: using a LBS multiple times

After describing how to provide geo-indistinguishability guarantees to users querying a LBS application a *single* time, we now discuss how to extend our solution to the case in which users wish to perform *multiple* queries.

In this scenario, the mechanism should protect multiple locations rather than one. But, what does it mean to enjoy privacy for multiple locations? As discussed in Section III-E, geo-indistinguishability can be naturally extended to this scenario. In short, the idea of being ℓ -private within r remains the same but for all locations simultaneously. In this way the locations, say, x_1, x_2 of a user employing the LBS twice remain indistinguishable from all pair of locations at (point-wise) distance at most r (ie, from all pairs x'_1, x'_2 such that $d(x_1, x'_1) \leq r$ and $d(x_2, x'_2) \leq r$).

A simple way of obtaining geo-indistinguishability guarantees when performing multiple queries is to employ our technique for protecting single locations to *independently* generate approximate locations for each of the user's locations. In this way, a user performing n queries via a mechanism providing ϵ -geo-indistinguishability enjoys $n\epsilon$ -geo-indistinguishability (see Theorem 3.2).

This solution might be satisfactory when the number of queries to perform remains fairly low, but in other cases impractical, due to the privacy degradation. It is worth noting that the canonical technique for achieving standard differential privacy (based on adding noise according to the Laplace distribution) suffers of the same privacy degradation problem (ϵ increases linearly on the number of queries). Several articles in the literature focus on this problem (see [20] for instance). We believe that the principles and techniques used to deal with this problem for standard differential privacy could be adapted to our scenario (either directly or motivationally). A fruitful direction to explore, in our particular scenario, is to employ the location history of the user together with the corresponding locations reported

to the LBS provider (ie, (x, z) pairs) to "adjust" the way approximate locations are generated (eg, report z whenever the user's location x' is nearby a location x that the mechanism has previously reported as z). This challenge constitutes our main focus for future work.

VI. SANITIZING DATASETS: US CENSUS CASE STUDY

In this section we present a sanitation algorithm for datasets containing geographical information. Roughly speaking, the algorithm iteratively sanitizes each of the geographic sensitive values in the dataset by means of the perturbation technique presented in Section IV.

A. The LODES dataset

We consider a realistic case study involving publicly available data developed by the U.S Census Bureau's Longitudinal Employer-Household Dynamics Program (LEHD). These data, called LEHD Origin-Destination Employment Statistics (LODES), are used by OnTheMap, a web-based interactive application developed by the US Census Bureau. The application enables, among other features, visualization of geographical information involving the residence and working location of US residents.

The LODES dataset includes information of the form $(hBlock, wBlock)$, where each pair represents a worker, the attribute $hBlock$ is the census block in which the worker lives, and $wBlock$ is the census block where the worker works. From this dataset it is possible to derive, by mapping home and work census blocks into their corresponding geographic centroids, a dataset with geographic information of the form $(hCoord, wCoord)$, where each of the coordinate pairs corresponds to a census block pair.

Due to privacy constraints and legal issues, data involving the residence location of individuals cannot be released without previous sanitation; thus, the LODES dataset is a sanitized version of the real data. However, for illustration purposes and wlog, in the remaining of this section we will treat the LODES dataset as if it were the real data. The Census Bureau uses a *synthetic data generation algorithm* [24], [12] to sanitize the LODES dataset. Roughly speaking, the algorithm interprets the dataset as an histogram where each $(hBlock, wBlock)$ pair is represented by a histogram bucket, the synthetic data generation algorithm sanitizes data by modifying the counts of the histogram. For that purpose, a statistical model is built from the LODES dataset and then a sanitized counterpart is obtained by sampling points from the model.

In the following we present a sanitizing algorithm for datasets with geographical information (eg, the LODES dataset) that provides formal privacy guarantees. In particular, our algorithm provides geo-indistinguishability guarantees under the assumption that the home census blocks values in the dataset are uncorrelated (see the discussion about uncorrelated points in Section III-E). Although this

assumption weakens the privacy guarantees provided by geo-indistinguishability, we believe that due to the anonymizing techniques applied by the Census Bureau to the released data involving census participants' information and to the large number of $(hCoord, work_coord)$ pairs within small areas contained in the dataset, a practical attack based on correlation of points is unlikely.

B. The Sanitizing Algorithm for a dataset of locations

Our sanitizing algorithm, described in Figure 13, takes as input (1) a dataset D to sanitize, (2) the privacy parameters ℓ and r (see Section III), and (3) the precision parameters u , v , δ_r and δ_θ , and the region \mathcal{A} . (see Section IV-B) and returns a sanitized counterpart of D . The algorithm is guaranteed to provide ℓ/r -geo-indistinguishability to the home coordinates of all individuals in the dataset (see discussion on protecting multiple locations in Section III-E).

We note that, in contrast to the approach used by the Census Bureau based on histogram's count perturbation, our algorithm modifies the geographical data itself (residence coordinates in this case). Therefore, our algorithm works at a more refined level than the synthetic data generation algorithm used by the Census Bureau; a less refined dataset can be easily obtained however – by just remapping each $(hCoord, wCoord)$ pair produced by our algorithm to its corresponding census block representation.

C. Experiments

In order to evaluate the accuracy of the sanitized dataset generated by our algorithm (and thus of our algorithm as a data sanitizer) we implemented our perturbation mechanism and conducted a series of experiments focusing on the “home-work commute distance” analysis provided by the OnTheMap application. This analysis provides, for a given area (specified as, say, state or county code), a histogram classifying the individuals in the dataset residing in the given area according to the distance between their residence location and their work location. The generated histogram contains four buckets representing different ranges of distance: (1) from zero to ten miles, (2) from ten to twenty five miles, (3) from twenty five to fifty miles, and (4) more than fifty miles.

Sanitizing Algorithm for a Dataset of Locations

Input: $D : hCoord \times wCoord$ // dataset to sanitize
 $\ell, r, u, v, \delta_r, \delta_\theta, \mathcal{A}$ // same as in Figure 9

Output: Sanitized version D' of input D

1. $D' = \emptyset$; // initializing output dataset
 2. $\epsilon = \ell/r$;
 3. **for each** $(c_h, c_w) \in D$ **do**
 4. $c'_h = \text{NoisyPt}(c_h, \epsilon, u, v, \delta_\theta, \delta_r, \mathcal{A})$; // sanitized point
 5. $D' = D' \cup \{(c'_h, c_w)\}$; // adding sanitized point
 6. **end-for**
 7. **return** D' ;
-

Figure 13. Our sanitizing algorithm, based on data perturbation

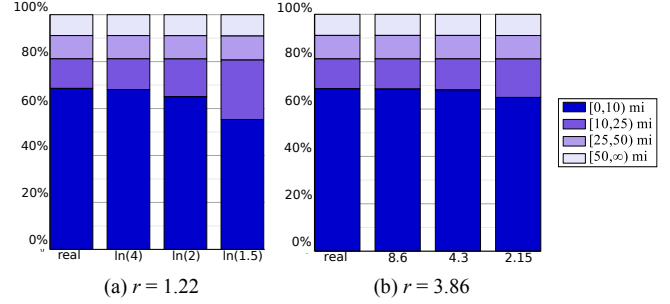


Figure 14. Home-work commute distance for various levels ℓ .

We have chosen the San Francisco (SF) County as residence area for our experimental analysis. Additionally, we restrict the work location of individuals residing in the San Francisco county to the state of California. The total number of individuals satisfying these conditions amounts to 374,390. All experiments have been carried on using version 6.0 of the LODES dataset. In addition, the mapping from census blocks to their corresponding centroids has been done using the 2011 TIGER census block shapefile information provided by the Census Bureau.

We now proceed to compare the LODES dataset – seen as a histogram – with several sanitized versions of it generated by our algorithm. Figure 14 (a) depicts how the geographical information degrades when fixing r to 1.22 miles (so to ensure geo-indistinguishability within 10% of the land area of the SF County) and varying ℓ . The precision parameters were chosen as follows: $u = 10^{-3}$ miles, \mathcal{A} 's diameter was set to 10^4 miles, and the standard double precision values for δ_r and δ_θ (for the corresponding ranges).

We have also conducted experiments varying r and fixing ℓ . For instance, if we want to provide geo-indistinguishability for 5%, 10%, and 25% of the land area of the SF county (approx. 46.87 mi²), we can set $r = 0.86$, 1.22, and 1.93 miles, respectively. Then by taking $\ell = \ln(2)$ we get an histogram very similar to the previous one. This is not surprising as the noise generated by our algorithm depends only on the ratios ℓ/r , which are similar for the values above.

As shown in Figure 14 (a), our algorithm has little effect on the bucket counts corresponding to mid/long distance commutes: over twenty five miles the counts of the sanitized dataset are almost identical to those of the input dataset – even for the higher degrees of privacy. For short commutes on the other hand, the increase in privacy degrades the accuracy of the sanitized dataset: several of the commutes that fall in the 0-to-10-miles bucket in the original data fall instead in the 10-to-25-miles bucket in the sanitized data.

After analyzing the accuracy of the sanitized datasets produced by our algorithm for several levels of privacy, we proceed to compare our approach with the one followed by the Census Bureau to sanitize the LODES dataset. Such comparison is unfortunately not straightforward; on the one hand, the approaches provide different privacy guarantees

(see discussion below) and, on the other hand, the Census Bureau is not able to provide us with a (sanitized) dataset sample produced by their algorithm (which would allow us to compare both approaches in terms of accuracy) as this might compromise the protection of the real data.

The algorithm used by the Census Bureau satisfies a notion of privacy that called (ϵ, δ) -probabilistic differential privacy, which is a relaxation of standard differential privacy that provides ϵ -differential privacy with probability at least $1 - \delta$ [12]. In particular, their algorithm satisfies $(8.6, 0.00001)$ -probabilistic differential privacy. This level of privacy could be compared to geo-indistinguishability for $\ell = 8.6$ and $r = 3.86$, which corresponds to providing protection in an area of the size of the SF County. Figure 14 (b) presents the results of our algorithm for such level of privacy and also for higher levels.

It becomes clear that, by allowing high values for ℓ ($\ell = 8.6 = \ln(5432)$, $\ell = 4.3 = \ln(74)$, and $\ell = 2.15 = \ln(9)$) it is possible to provide privacy in large areas without significantly diminishing the quality of the sanitized dataset.

VII. RELATED WORK

Much of the related work has been already discussed in Section II, here we only mention the works that were not reported there. We refer to [25] for an excellent survey on privacy methods for geolocation.

LISA [26] provides location privacy by preventing an attacker from relating any particular point of interest (POI) to the user's location. That way, the attacker cannot infer which POI the user will visit next. The privacy metric used in this work is *m-unobservability*. The method achieves *m-unobservability* if, with high probability, the attacker cannot relate the estimated location to at least m different POIs in the proximity.

SpaceTwist [27] reports a fake location (called the “anchor”) and queries the geolocation system server incrementally for the nearest neighbors of this fake location until the k -nearest neighbors of the real location are obtained.

VIII. CONCLUSION AND FUTURE WORK

In this paper we have presented a framework for achieving privacy in location-based applications, taking into account the desired level of protection as well as the side-information that the attacker might have. The core of our proposal is a new notion of privacy, that we call geo-indistinguishability, and a method, based on a bivariate version of the Laplace function, to perturbate the actual location. We have put a strong emphasis in the formal treatment of the privacy guarantees, both in giving a rigorous definition of geo-indistinguishability, and in providing a mathematical proof that our method satisfies such property. We also have shown how geo-indistinguishability relates to the popular notion of

differential privacy. Finally, we have illustrated the applicability of our method with two case studies: interaction with a POI-retrieval service, and sanitization of the LODS dataset.

In the future we aim at extending our method to cope with more complex applications, possibly involving the sanitization of several (potentially related) locations. One important aspect to consider when generating noise on several data is the fact that their correlation may degrade the level of protection. We aim at devising techniques to control the possible loss of privacy and to allow the composability of our method.

In a recent paper [28] it has been shown that, due to finite precision and rounding effects of floating-point operations, the standard implementations of the Laplacian mechanism result in an irregular distribution which causes the loss of the property of differential privacy. The same paper proposes a solution based on a post-processing snapping procedure. In our setting, we suspect that we encounter the same kind of problem when we draw the radius according to the bivariate Laplacian. We believe that the solution proposed in [28] applies also to our case, and that the snapping procedure will cause an effect equivalent to a loss of precision. This means that, even in a double-precision machine, the remapping from polar to cartesian coordinates may require a non-negligible additional amount of noise in order to preserve differential privacy, i.e. the gap between ϵ and ϵ' (cf. Figure 7) may become larger. We plan to investigate this relation more accurately in the future.

REFERENCES

- [1] Pew Internet & American Life Project. www.pewinternet.org.
- [2] J. E. Dobson and P. F. Fisher, “Geoslavery,” *Technology and Society Magazine, IEEE*, vol. 22, no. 1, pp. 47–52, 2003.
- [3] C. Dwork, “Differential privacy,” in *Proc. of ICALP*, ser. LNCS, vol. 4052. Springer, 2006, pp. 1–12.
- [4] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking,” in *Proc. of MobiSys*. USENIX, 2003.
- [5] B. Gedik and L. Liu, “Location privacy in mobile systems: A personalized anonymization model,” in *Proc. of ICDCS*. IEEE, 2005, pp. 620–629.
- [6] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, “The new casper: Query processing for location services without compromising privacy,” in *Proc. of VLDB*. ACM, 2006, pp. 763–774.
- [7] H. Kido, Y. Yanagisawa, and T. Satoh, “Protection of location privacy using dummies for location-based services,” in *Proc. of ICDE Workshops*, 2005, p. 1248.
- [8] P. Shankar, V. Ganapathy, and L. Iftode, “Privately querying location-based services with sybilquery,” in *Proc. of UbiComp*. ACM, 2009, pp. 31–40.
- [9] B. Bamba, L. Liu, P. Pesti, and T. Wang, “Supporting anonymous location queries in mobile environments with privacygrid,” in *Proc. of WWW*. ACM, 2008, pp. 237–246.

- [10] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in *Proc. of PERSASIVE*, ser. LNCS, vol. 3468. Springer, 2005, pp. 152–170.
- [11] M. Xue, P. Kalnis, and H. Pung, “Location diversity: Enhanced privacy protection in location based services,” in *Proc. of LoCA*, ser. LNCS, vol. 5561. Springer, 2009, pp. 70–87.
- [12] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber, “Privacy: Theory meets practice on the map,” in *Proc. of ICDE*. IEEE, 2008, pp. 277–286.
- [13] S.-S. Ho and S. Ruan, “Differential privacy for location pattern mining,” in *Proc. of SPRINGL*. ACM, 2011, pp. 17–24.
- [14] A. Khoshgozaran and C. Shahabi, “Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy,” in *Proc. of SSTO*, ser. LNCS, vol. 4605. Springer, 2007, pp. 239–257.
- [15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: anonymizers are not necessary,” in *Proc. of SIGMOD*. ACM, 2008, pp. 121–132.
- [16] J. Reed and B. C. Pierce, “Distance makes the types grow stronger: a calculus for differential privacy,” in *Proc. of ICFP*. ACM, 2010, pp. 157–168.
- [17] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, “Fairness through awareness,” in *Proc. of ITCS*. ACM, 2012, pp. 214–226.
- [18] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, “Enhancing differential privacy: from hamming to general metrics,” Tech. Rep., 2012, <http://www.lix.polytechnique.fr/~catuscia/papers/DifferentialPrivacy/extDP.pdf>.
- [19] A. Blum, K. Ligett, and A. Roth, “A learning theory approach to non-interactive database privacy,” in *Proc. of STOC*. ACM, 2008, pp. 609–618.
- [20] A. Roth and T. Roughgarden, “Interactive privacy via the median mechanism,” in *Proc. of STOC*, 2010, pp. 765–774.
- [21] C. Dwork, “A firm foundation for private data analysis,” *Communications of the ACM*, vol. 54, no. 1, pp. 86–96, 2011.
- [22] K. Lange and J. S. Sinsheimer, “Normal/independent distributions and their applications in robust regression,” *J. of Comp. and Graphical Statistics*, vol. 2, no. 2, pp. 175–198, 1993.
- [23] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proc. of TCC*, ser. LNCS, vol. 3876. Springer, 2006, pp. 265–284.
- [24] D. B. Rubin, “Discussion: Statistical disclosure limitation,” *Journal of Official Statistics*, vol. 9, no. 2, pp. 461–468, 1993.
- [25] M. Terrovitis, “Privacy preservation in the dissemination of location data,” *SIGKDD Explorations*, vol. 13, no. 1, pp. 6–18, 2011.
- [26] Z. Chen, “Energy-efficient information collection and dissemination in wireless sensor networks.” Ph.D. dissertation, University of Michigan, 2009.
- [27] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, “Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services,” in *Proc. of ICDE*. IEEE, 2008, pp. 366–375.
- [28] I. Mironov, “On significance of the least significant bits for

differential privacy,” in *Proc. of CCS*. ACM, 2012, pp. 650–661.

APPENDIX

In this appendix we provide the technical details that have been omitted from the main body of the paper.

A. Results from Section III

Theorem 3.1: Geo-indistinguishability-I, II, III coincide.

Proof: The equivalence of Geo-indistinguishability-I and III can be shown by applying Bayes’ law. We show here the equivalence between Geo-indistinguishability-II and III.

Assume that K satisfies geo-indistinguishability-III. We first show that for all $r > 0$:

$$\begin{aligned}
 P(S|B_r(x)) &= \sum_{x' \in B_r(x)} P(S, x'|B_r(x)) \\
 &= \sum_{x' \in B_r(x)} P_X(x'|B_r(x)) K(x')(S) \\
 &\geq \sum_{x' \in B_r(x)} P_X(x'|B_r(x)) e^{-\epsilon r} K(x)(S) \quad d(x, x') \leq r \\
 &= e^{-\epsilon r} K(x)(S)
 \end{aligned}$$

Then

$$P(x|S, B_r(x)) = \frac{P(S|x)}{P(S|B_r(x))} P(x|B_r(x)) \leq e^{\epsilon r} P(x|B_r(x))$$

For the opposite direction, let $x_1, x_2 \in \mathcal{X}$, let $r = d(x_1, x_2)$ and define a prior distribution $P_X^t(x)$ as:

$$P_X^t(x) = \begin{cases} t & x = x_1 \\ 1 - t & x = x_2 \\ 0 & \text{otherwise} \end{cases}$$

Using that prior for $t \in (0, 1)$ we have for all S :

$$\begin{aligned}
 K(x_1)(S) &= P(S|x_1) \\
 &= P(S|x_1, B_r(x_1)) \\
 &= \frac{P(x_1|S, B_r(x_1))}{P(x_1|B_r(x_1))} P(S|B_r(x_1)) \\
 &\leq e^{\epsilon r} P(S|B_r(x_1)) \\
 &\leq e^{\epsilon r} \sum_{x \in \mathcal{X}} P(S, x|B_r(x_1)) \\
 &\leq e^{\epsilon r} (tP(S|x_1) + (1 - t)P(S|x_2)) \\
 &\leq e^{\epsilon r} (tK(x_1)(S) + (1 - t)K(x_2)(S))
 \end{aligned}$$

Note that we need $t \in (0, 1)$ so that $P_X^t(x_1), P_X^t(x_2)$ are positive and the conditional probabilities can be defined. Finally, taking the $\lim_{t \rightarrow 0}$ on both sides of the above inequality we get $K(x_1)(S) \leq e^{\epsilon r} K(x_2)(S)$ ■

Theorem 3.2: If K_0 satisfies ϵ -geo-indistinguishability, then \mathcal{K} satisfies $n\epsilon$ -geo-indistinguishability.

Proof: Let $\mathbf{x} = (x_1, \dots, x_n), \mathbf{x}' = (x'_1, \dots, x'_n)$ such that $d_\infty(\mathbf{x}, \mathbf{x}') \leq r$. This implies that $d(x_i, x'_i) \leq r$, $1 \leq i \leq n$. We have:

$$\begin{aligned} P(\mathbf{z}|\mathbf{x}) &= \prod_i P(z_i|x_i) \\ &\leq \prod_i e^{\epsilon r} P(z_i|x'_i) \\ &= e^{n\epsilon r} \prod_i P(z_i|x'_i) \\ &= e^{n\epsilon r} P(\mathbf{z}|\mathbf{x}') \end{aligned}$$

■

B. The planar laplacian satisfies geo-indistinguishability

Given the definition of $D_\epsilon(x_0)(x)$ in (1), by triangular inequality we have

$$D_\epsilon(x_0)(x) \leq e^{\epsilon d(x_0, x'_0)} D_\epsilon(x'_0)(x)$$

Using well-known properties of integrals, we derive

$$\int_S D_\epsilon(x_0)(x) ds \leq \int_S e^{\epsilon d(x_0, x'_0)} D_\epsilon(x'_0)(x) ds$$

and

$$\int_S D_\epsilon(x_0)(x) ds \leq e^{\epsilon d(x_0, x'_0)} \int_S D_\epsilon(x'_0)(x) ds$$

Now, taking into account the definition of K :

$$K(x_0)(S) = \int_S D_\epsilon(x_0)(x) ds$$

we derive

$$K(x_0)(S) \leq e^{\epsilon d(x_0, x'_0)} K(x'_0)(S)$$

■

C. The discretization preserves geo-indistinguishability

Theorem 4.1: Assume $r_{\max} < u/\delta_\theta$, and let $q = u/r_{\max}\delta_\theta$. Let $\epsilon, \epsilon' \in \mathbb{R}^+$ such that

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \leq \epsilon$$

Then $K_{\epsilon'}$ provides ϵ -geo-indistinguishability within the range of r_{\max} . Namely, if $d(x_0, x), d(x'_0, x) \leq r_{\max}$ then:

$$K_{\epsilon'}(x_0)(x) \leq e^{\epsilon d(x_0, x'_0)} K_{\epsilon'}(x'_0)(x).$$

Proof: The case in which $x_0 = x'_0$ is trivial. We consider therefore only the case in which $x_0 \neq x'_0$. Note that in this case $d(x_0, x'_0) \geq u$. We proceed by determining an upper bound on $K_{\epsilon'}(x_0)(x)$ and a lower bound on $K_{\epsilon'}(x'_0)(x)$ for generic x_0, x'_0 and x such that $d(x_0, x), d(x'_0, x) \leq r_{\max}$. Let S be the set of points for which x is the closest point in \mathcal{G} , namely:

$$S = R(x) = \{y \in \mathbb{R}^2 \mid \forall x' \in \mathcal{G}. d(y, x') \leq d(y, x')\}$$

Ideally, the points remapped in x would be exactly those in S . However, due to the finite precision of the machine, the points actually remapped in x are those of $R_{\mathcal{W}}(x)$ (see Section IV-B). Hence the probability of x is that of S plus

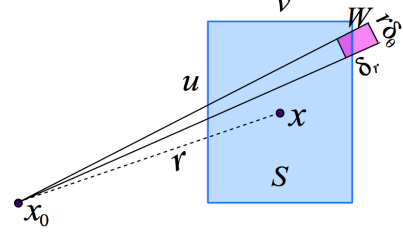


Figure 15. Bounding the probability of x in the discrete Laplacian.

or minus the small rectangles⁶ W of size $\delta_r \times r\delta_\theta$ at the border of S , where $r = d(x_0, x)$, see Figure 15. Let us denote by S_W the total area of these small rectangles W on one of the sides of S . Since $d(x_0, x) \leq r_{\max} < u/\delta_\theta$, and $\delta_r < r_{\max}\delta_\theta$, we have that S_W is less than $1/q$ of the area of S , where $q = u/r_{\max}\delta_\theta$. The probability density on this area differs at most by a factor $e^{\epsilon' u}$ from that of the other points in S . Finally, note that on two sides of S the rectangles W contribute positively to $K_{\epsilon'}(x_0)(x)$, while on two sides they contribute negatively. Summarizing, we have:

$$K_{\epsilon'}(x_0)(x) \leq (1 + \frac{2e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x_0)(x_1) ds \quad (3)$$

and

$$(1 - \frac{2e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x'_0)(x_1) ds \leq K_{\epsilon'}(x'_0)(x) \quad (4)$$

Observe now that

$$\frac{D_{\epsilon'}(x_0)(x_1)}{D_{\epsilon'}(x'_0)(x_1)} = e^{-\epsilon' (d(x_0, x_1) - d(x'_0, x_1))}$$

By triangular inequality we obtain

$$D_{\epsilon'}(x_0)(x_1) \leq e^{\epsilon' d(x_0, x'_0)} D_{\epsilon'}(x'_0)(x_1)$$

from which we derive

$$\int_S D_{\epsilon'}(x_0)(x_1) ds \leq e^{\epsilon' d(x_0, x'_0)} \int_S D_{\epsilon'}(x'_0)(x_1) ds \quad (5)$$

from which, using (3), (5), and (4), we obtain

$$K_{\epsilon'}(x_0)(x) \leq e^{\epsilon' d(x_0, x'_0)} K_{\epsilon'}(x'_0)(x) \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \quad (6)$$

Assume now that

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \leq \epsilon$$

Since we are assuming $d(x_0, x'_0) \geq u$, we derive:

$$e^{\epsilon' d(x_0, x'_0)} \frac{q + 2e^{\epsilon' u}}{q - 2e^{\epsilon' u}} \leq e^{\epsilon d(x_0, x'_0)} \quad (7)$$

Finally, from (6) and (7), we conclude. ■

⁶ W is actually a fragment of a circular crown, but since δ_θ is very small, it approximates a rectangle. Also, the side of W is not exactly $r\delta_\theta$, it is a number in the interval $[(r - u/\sqrt{2})\delta_\theta, (r + u/\sqrt{2})\delta_\theta]$. However $u/\sqrt{2}\delta_\theta$ is very small with respect to the other quantities involved, hence we consider negligible this difference.

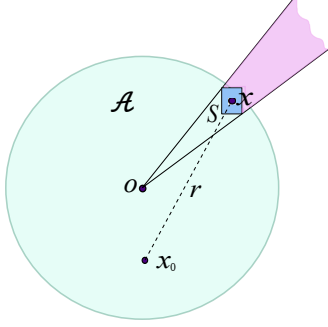


Figure 16. Probability of x in the truncated discrete laplacian.

D. The truncation preserves geo-indistinguishability

Theorem 4.2: Let r_{\max} , ϵ and ϵ' satisfy the premise of Theorem 4.1. If $r_{\max} \geq \text{diam}(\mathcal{A})$, then $K_{\epsilon'}^T$ provides ϵ -geo-indistinguishability within \mathcal{A} .

Proof: The proof proceeds like the one for Theorem 4.1, except when $R(x)$ is on the border of \mathcal{A} . In this latter case, the probability on x is given not only by the probability on $R(x)$ (plus or minus the small rectangles W – see the proof of Theorem 4.1), but also by the probability of the part C of the cone determined by o , $R(x)$, and lying outside \mathcal{A} (see Figure 16). Following a similar reasoning as in the proof of Theorem 4.1 we get

$$K_{\epsilon'}^T(x_0)(x) \leq \left(1 + \frac{2e^{\epsilon' u}}{q}\right) \int_{S \cup C} D_{\epsilon'}(x_0)(x_1) ds$$

and

$$\left(1 - \frac{2e^{\epsilon' u}}{q}\right) \int_{S \cup C} D_{\epsilon'}(x'_0)(x_1) ds \leq K_{\epsilon'}^T(x'_0)(x)$$

The rest follows as in the proof of Theorem 4.1. ■